UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,774	01/03/2002	Arch D. Robison	42390P13132	7002

8791 7590 10/23/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
FOWLKES, ANDRE R	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 10/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/037,774	ROBISON, ARCH D.	
	Examiner	Art Unit	
	Andre R. Fowlkes	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 August 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 8/7/06 has been entered.

2. Claims 1-28 are pending. Claims 1, 4-5, 7-8, 11, -12, 15, 17-19, 22, 24 and 28 have been amended.

Claim Rejections - 35 USC § 112

3. The rejection of claims 1-28 under 35 U.S.C. 112, first paragraph, is withdrawn, in view of applicant's arguments.

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-6, 15-21 and 22-28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 8, 21, 22 and 25 recites the limitation "executable instructions" (e.g. claim 1 lines 9 and 10, and claim 15 lines 1-2.) There is insufficient antecedent basis for this limitation in the claim. The rejection of base claims, 1, 8 and 22, is necessarily incorporated into the dependent claims.

Specification

6. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The following title is suggested: System and Method to Reduce the Size of Source Code in a Processing System.

7. The disclosure is objected to because of the following informalities: "executable instructions/code" should be –source instructions—in the entire specification, e.g. paragraph 1 line 1.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-28 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Wulf, et al, (Wulf), The Design of an Optimizing Compiler, American Elsevier Publishing Co., Inc. 1975, ISBN: 0-444-00158-1 (art made of record).

As per claim 1, Wulf discloses a method comprising:

- identifying a plurality of fork subgraph structures within a graph structure constructed for a plurality of source code instructions (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter"),

- identifying, prior to register allocation, a plurality of unifiable variables within each fork subgraph structure of said plurality of fork subgraph structures, which are not simultaneously used in said plurality of source code instructions (p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these

sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)"),

- constructing a dependence graph of said plurality of source code instructions (p. 6:21-26, "a tree representation (i.e. data dependence graph) of the parsed program unit (i.e. source code instructions) and a set of lists, generally threads running through the tree, which define feasible global optimizations"),

using said dependence graph to identify at least one unifiable instruction of said plurality of executable instructions, within said plurality of fork subgraph structures (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these sets represent computations which may

under certain circumstances be moved forward to the end of the linear block (i.e. the handle)"),

- transferring at least one unifiable instruction of said plurality of executable instructions from a fork of a corresponding fork subgraph structure of said plurality of fork subgraph structures to a handle of said corresponding fork subgraph structure; said at least one unifiable instruction containing at least one unifiable variable of said plurality of unifiable variables; and unifying each unifiable variable within said at least one unifiable instruction (p. 29:29-31, "The linear blocks alpha and omega contain those expressions factored forward and backward from all of the branches, B.sub.j", and figures 6-7 and associated text, describe transferring a unifiable instruction from a fork of a corresponding fork subgraph structure of said plurality of fork subgraph structures (Fig. 6 B.sub.1 to B.sub.n) to a handle (Fig 7. E.prime.sub1) of said corresponding fork subgraph structure; said at least one unifiable instruction containing at least one unifiable variable of said plurality of unifiable variables; and unifying each unifiable variable within said at least one unifiable instruction).

As per claim 2, the rejection of claim 1 is incorporated and further, Wulf discloses that identifying said plurality of unifiable variables further comprises:

- constructing an interference graph structure for a plurality of local variables within said each fork subgraph structure (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code)

language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter"),

- said plurality of local variables including said plurality of unifiable variables (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)"),

- identifying said plurality of unifiable variables as variables having non-overlapping live ranges within said interference graph structure (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin

by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)").

As per claim 3, the rejection of claim 2 is incorporated and further, Wulf discloses that **said interference graph structure indicates which variables of said plurality of local variables are simultaneously used in said plurality of source code instructions and cannot be unified** (p. 15:24-26, "Each type of tree (graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is

the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)").

As per claim 4, the rejection of claim 1 is incorporated and further, Wulf discloses that identifying said plurality of unifiable variables further comprises: **constructing a data dependence graph structure for said plurality of source code instructions; and identifying said plurality of unifiable variables using said data dependence** analysis (p. 15:24-26, "Each type of tree (data dependence graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of

output variables", and p. 29:15-20, "the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the elements of these sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)").

As per claim 5, the rejection of claim 1 is incorporated and further, Wulf discloses **initializing a flag for said at least one unifiable instruction** (p. 15:24-26, "Each type of tree (data dependence graph structure) node is uniquely associated with some delimiter in the source (code) language, e.g. the node representing a conditional expression (i.e. fork subgraph structure) is associated with the *if* delimiter", and p. 22:6-20, "We begin by considering the ordering relations inherent in a representation of a program P. There are several: the lexical order..., the precedence-induced order of evaluation, both data-sensitive and data-insensitive order induced by control flow, and so forth. Two such orderings are important... The first is the order that results from considering a program (or subgraph structure of a representation of a program) as a mapping from its set of input variables to its set of output variables", and p. 29:15-20, "(flagging) the expressions in epilog(B) and postlog(B) have no epi-dominators or post-dominators (i.e. the variables in the set (epilog(B) U postlog(B)) are unifiable variables that are not simultaneously used in this code section), respectively in B. Thus the

elements of these sets represent computations which may under certain circumstances be moved forward to the end of the linear block (i.e. the handle)").

As per claim 6, the rejection of claim 5 is incorporated and further, Wulf discloses **removing said at least one unifiable instruction from subsequent forks of said corresponding fork subgraph structure** ((p. 29:29-31, "The linear blocks alpha and omega contain those expressions factored forward and backward from all of the branches, B.sub.j", and figures 6-7 and associated text, describe transferring a unifiable instruction from a fork of a corresponding fork subgraph structure of said plurality of fork subgraph structures (Fig. 6 B.sub.1 to B.sub.n) to a handle (Fig 7. E.prime.sub1) of said corresponding fork subgraph structure; said at least one unifiable instruction containing at least one unifiable variable of said plurality of unifiable variables; and unifying each unifiable variable within said at least one unifiable instruction).

As per claim 7, the rejection of claim 4 is incorporated and further, Wulf discloses that **said data dependence graph structure contains a plurality of dependence arcs, each dependence arc connecting two instructions of said plurality of source code instructions contained within said fork of said corresponding fork subgraph structure** (p. 6:21-26, "a tree representation (i.e. data dependence graph) of the parsed program unit (i.e. source code instructions) and a set of lists, generally threads running through the tree, which define feasible global optimizations").

As per claims 8-14, this is a system version of the claimed method discussed above, in claims 1-7, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Wulf's discussion of the Bliss/11 compiler, e.g. 6:21-29:31.

As per claims 15-21, this is a computer readable medium version of the claimed method discussed above, in claims 1-7, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Wulf's discussion of the Bliss/11 compiler, e.g. 6:21-29:31.

As per claims 22-28, this is another system version of the claimed method discussed above, in claims 1-7, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Wulf's discussion of the Bliss/11 compiler, e.g. 6:21-29:31.

Response to Arguments

10. Applicant's arguments with respect to claims 1-28 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571)

272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ARF

TUAN DAM
SUPERVISORY PATENT EXAMINER